DUE: Wednesday, February 13, 2:20PM in class.

You may discuss this assignment with whomever you wish, but please submit work in teams of ***two to three*** people. In particular, teams are encouraged to discuss the assignment with other teams, but please submit one assignment per team. Please, no individual submissions. Please indicate all team member names on the submission. Please prepare solutions in a ***concise and organized fashion***. I prefer typeset presentations (e.g., cut and paste code/output into MS Word with added exposition when appropriate; knitr/R Markdown via Rstudio, etc.)—probably most appropriate for presenting (fixed width font) code/output, at least—but neatly handwritten presentations may also be appropriate for some problems. Sloppily prepared/disorganized solutions will not receive full credit.
NOTE: You will be asked to provide peer assessment for each of your team members (excluding yourself!) via a score from 0 to 10, 10 being best. Share these assessments only with me, please. Please use a standard 8.5x11 inch sheet of paper for your scores.

For this homework, we will work with data that I obtained from [MR07, Chap. 3] (who obtained it elsewhere). The data consist of measurements associated with processionary caterpillars. The observed output, y, is the (natural log of the) average number of caterpillar nests per tree in 500 m^2 of forest, and is in the *last* column (11) of the data set. The first ten columns consist of corresponding observations of $k = 10$ input variables, roughly described in the following table.

x_1	elevation (m)
x_2	slope (degrees)
x_3	number of pines in the area
x_4	representative tree height (m)
x_5	representative tree diameter
x_6	settlement density index
x_7	site orientation index
x_8	height of dominant tree (m)
x_9	vegetation strata index
x_{10}	settlement mix index

NOTE: You may have to download/install an `R` package, which I assume you can do.

1. The data are available in BbLearn in the file, `caterpillar.txt`. Read the caterpillar data set into `R` and use `R` to: (i) name the columns of the resulting data frame as `y, x1,...,x10`, and (ii) display the first 3 and last 3 records of your data set. Show your code and output.
   ANS:

```
> cat.df<- read.table(file="../../Spring2018/datasets/caterpillar.txt")
> shortnames<- c(paste0("x",1:10),"y")
```

```
> names(cat.df)<- shortnames
> head(cat.df,n=3); tail(cat.df,n=3)

    x1 x2 x3  x4   x5  x6  x7  x8  x9 x10    y
1 1200 22  1 4.0 14.8 1.0 1.1 5.9 1.4 1.4 2.37
2 1342 28  8 4.4 18.0 1.5 1.5 6.4 1.7 1.7 1.47
3 1231 28  5 2.4  7.8 1.3 1.6 4.3 1.5 1.4 1.13
     x1 x2 x3  x4   x5  x6  x7   x8  x9 x10    y
31 1228 31  6 5.4 21.8 1.3 1.7  7.0 1.5 1.9 0.35
32 1229 21 11 5.8 16.7 1.7 1.8 10.0 2.3 2.0 0.21
33 1310 36 17 5.2 17.8 2.3 1.9 10.3 2.6 2.0 0.03
```

2. Create an appropriate scatter plot matrix, similar to that seen in our notes. You may use the `pairs` function or a similar function in `R`. Use the `labels` option of the `pairs` function to provide more informative variable names than given above. If you use another plot function, label your plot in a similar manner. Show your code and plot.
ANS:

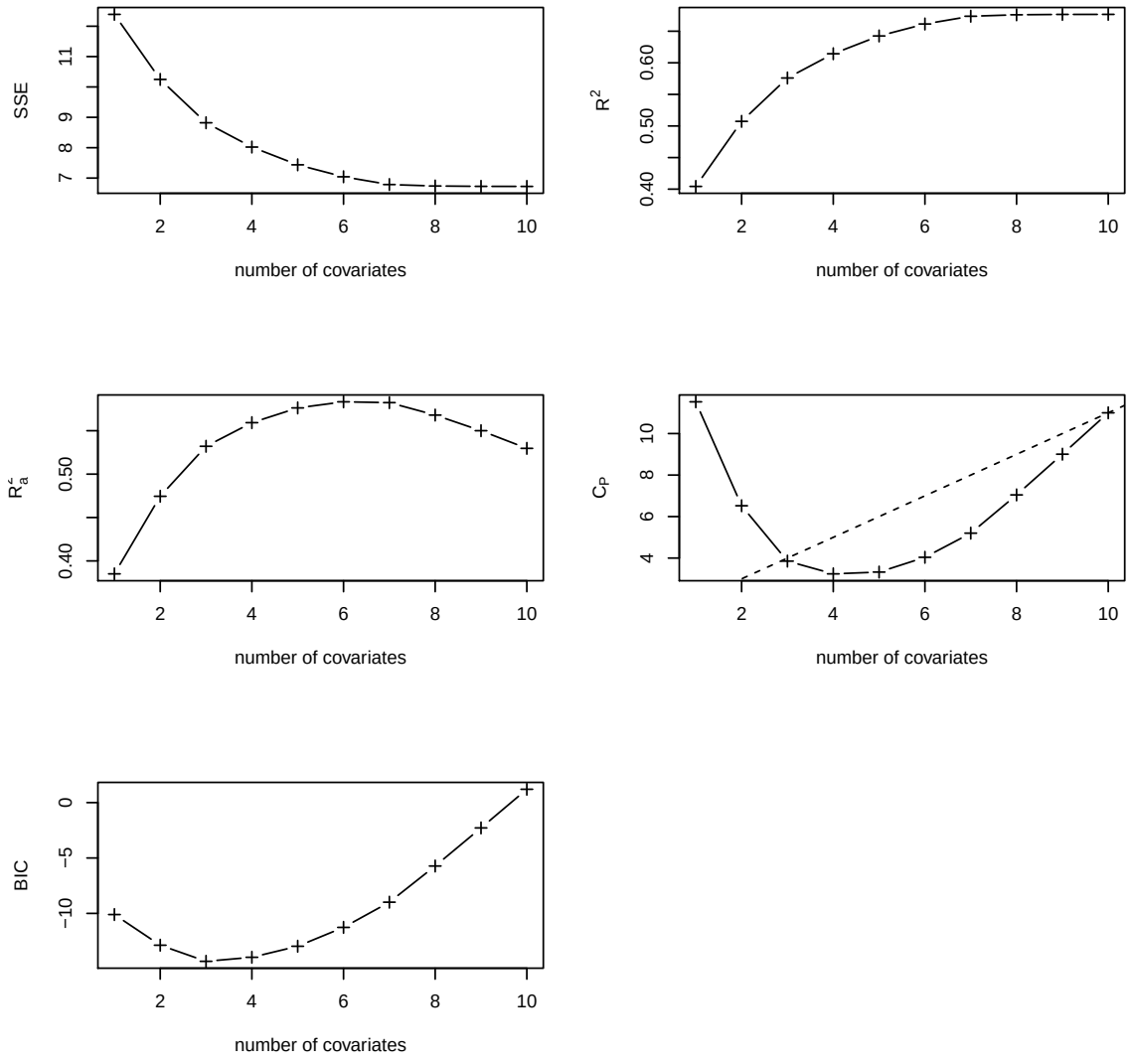
```
> longnames<- c("elev","slope","npines","reptreeht","reptreedia",
+               "densndx","orientndx","domtreeht","vegndx","mixndx",
+               "lnumcat")
> pairs(y ~ . ,
+       upper.panel = panel.smooth,
+       lower.panel = NULL,
+       data= cat.df,
+       labels=longnames[c(11,1:10)])
```

3. How many linear models are possible using all possible subsets of the inputs, including the model consisting only of an intercept?
   ANS:

```
> 2^10

[1] 1024

> sum(choose(10,0:10))

[1] 1024
```

4. Use the function, `regsubsets`, in the `leaps` package, to perform an exhaustive search for the best model/subset. In particular, create a matrix of plots, one for each of SSE, R^2, R_a^2, C_P, and BIC, each plot showing the *single* best model for each of the possible sizes. For the plot of C_P vs. number of covariates, add the reference line, $C_P = 1 + |P|$. Show your code and plots, appropriately annotated.
ANS:

```
> library(leaps)
> k<-10
> cat.reg.sum<- summary(cat.reg<-
+                           regsubsets(y ~ .,
+                                      data=cat.df,
+                                      nvmax=10))
> ## summary(p1best.rsub) ## not run
> par(mfrow=c(3,2))
> plot(y=cat.reg.sum$rss, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(SSE),
+      pch=3, type="b")
> plot(y=cat.reg.sum$rsq, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(R^2),
+      pch=3, type="b")
> plot(y=cat.reg.sum$adjr2, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(R[a]^2),
+      pch=3, type="b")
> plot(y=cat.reg.sum$cp, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(C[P]),
+      pch=3, type="b")
> abline(c(1,1), lty=2)
> plot(y=cat.reg.sum$bic, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(BIC),
+      pch=3, type="b")
> par(mfrow=c(1,1))
```



5. Identify the best model(s), over all sizes, as selected by each of the criteria, BIC, C_P, and R^2_a; this will result in 1, 2, or 3 models, depending on how these criteria (dis)agree. **ANS**: (See answer for 8 for discussion)

```
> (best.bic<- which.min(cat.reg.sum$bic))

[1] 3

> (best.cp1<- which.min(cat.reg.sum$cp))

[1] 4
```

```
> best.cp2<- 3
> (best.adjr2<- which.max(cat.reg.sum$adjr2))

[1] 6

> best.bhat.list<- coef(cat.reg.sum$obj, id=1:k)[c(best.bic,
+                                                  best.cp1,
+                                                  best.cp2,
+                                                  best.adjr2)]
> names(best.bhat.list)<- c("bic","cp1","cp2","adjr2")
> best.bhat.list

$bic
 (Intercept)           x1           x2           x9
 5.711169486 -0.002148421 -0.030582445 -0.598566825

$cp1
 (Intercept)           x1           x2           x6           x9
 6.823110029 -0.002854864 -0.030868136  0.598591740 -1.227515259

$cp2
 (Intercept)           x1           x2           x9
 5.711169486 -0.002148421 -0.030582445 -0.598566825

$adjr2
 (Intercept)           x1           x2           x4           x5
 6.764042285 -0.002787789 -0.033584770 -0.450509338  0.107717534
          x6           x9
 0.434039759 -0.870773297
```

6. Why does it not seem plausible to use the validation set approach, using MSPR, for this data set? Please be concise.
ANS: We have $n = 33$ observations. Short of obtaining another validation set, indepedent of the current training set, we might (randomly) split the training set into, e.g., $n = 17$ and $n^* = 16$ observations for independent training and validation sets, respectively. But, we are estimating up to $k + 1 = 11$ parameter models, which would mean we have about 1.5 observations per parameter to train models, which would make for relatively variable estimators, $\widehat{\boldsymbol{\beta}}$ and hence variable fitted models $\widehat{\mu}$. And, we would predict on a small validation set making MSPR a relatively poor estimate of generalization error. Thus, we seek alternatives to the validation set approach, e.g., information criteria such as AIC, BIC, etc., and cross-validation, none of which require

an independently collected validation set.

7. We only briefly introduced K-fold cross-validation in our notes as a more popular and modern (ML) alternative to the validation set approach to estimate generalization error and select models. Here, we conduct K-fold CV using $K = n$ (n-fold CV, or CV(n), or LOOCV). For this homework, we perform n-fold CV. That is, $K = n$ folds produce n training sets of size $n_i = n - 1$ to predict on each of the corresponding n validation sets of size $n_i^* = 1$.

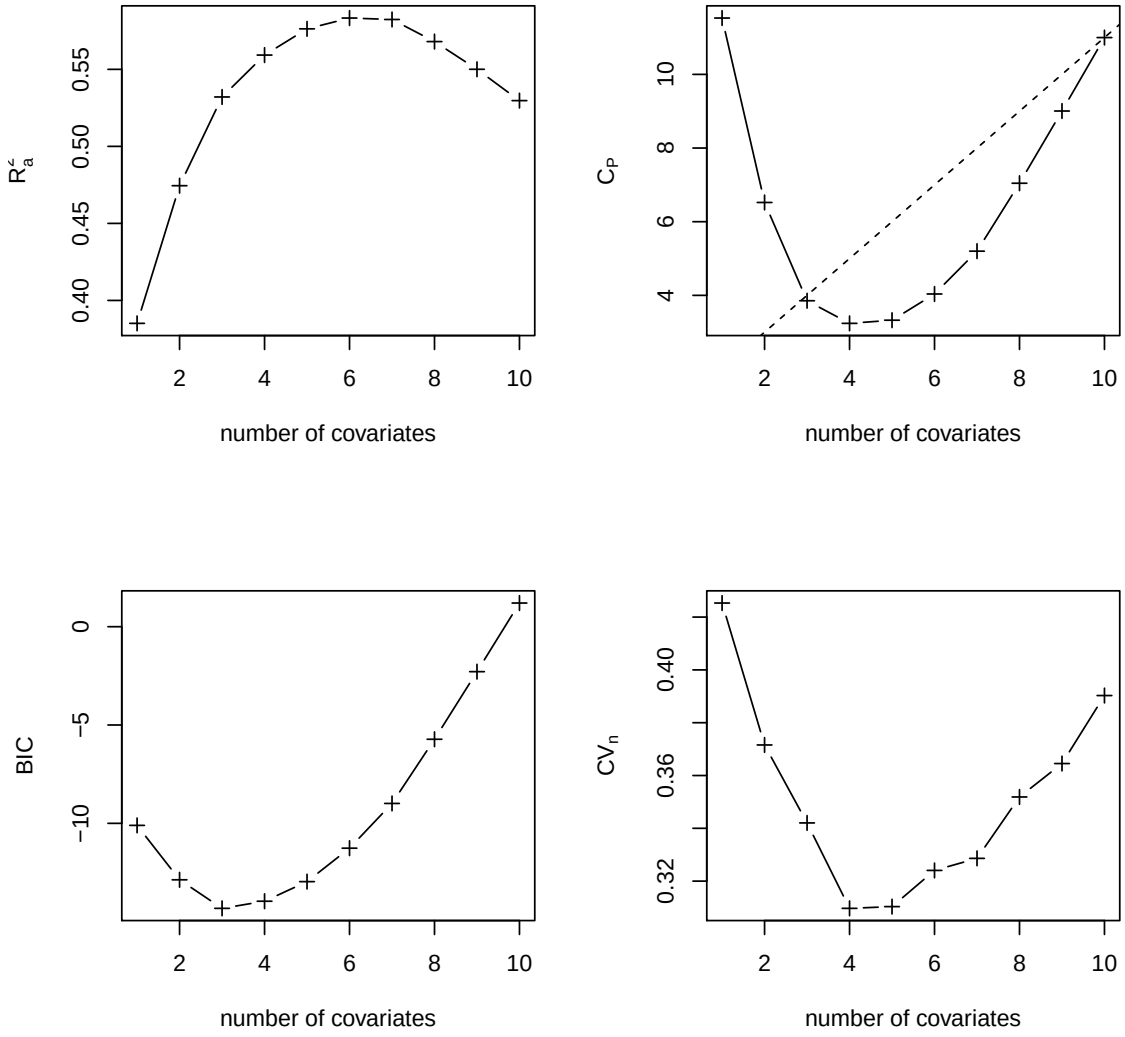
   We'll leave the details to the function, `cv.glm`, in the `boot` package. To use this function, we need to fit our model(s) using the function, `glm`, in the `stats` package. For example, the code, below, uses `glm` to fit a linear model, just as `lm` does, and just as `regsubsets` does, above. The code then performs n-fold CV ($K = n$ by default) using `cv.glm`. CV(K=n), as we have defined it, is contained in the first element of the list component named `delta` in the list object, `eg.cv.n`, returned by `cv.glm`.

```
> eg.glm<- glm(y ~ x1 + x2 + x3, data=cat.df)
> library(boot)
> eg.cv.n<- cv.glm(data=cat.df, glmfit=eg.glm)
> (cv.n<- eg.cv.n$delta[1]) ## CV(n)


[1] 0.426571
```

   Using the above code to get you started, perform CV(n) on the best fitted model of each size as indicated by the regsubsets procedure, above. You will have to figure out what these models are (you should already know 1-3 of these, by previous work), fit each one using **glm**, then compute CV(n) for each one. Prepare a plot of CV(n) vs model size, similar to the above plots using other criteria. (NOTE: We just consider these few fitted models for which to compute CV(n); we do not compute CV(n) for all the models that you counted in problem 3.)
   ANS: (See answer for 8 for discussion.)

```
> ## Overwriting previous list of same name:
> best.bhat.list<- coef(cat.reg.sum$obj, id=1:k)
> ## BTW, see help(paste) and examples in help(formula):
> best.glm.cvn<- sapply(best.bhat.list, FUN=function(x,dat){
+     fmla<- as.formula(paste("y ~ ", paste(names(x)[-1], collapse= "+")))
+     fit<-glm(fmla, data=dat)
+     cv.n<-cv.glm(data=dat, glmfit=fit)
+     return(list("fit"=fit, "cv.n"=cv.n))
+     },
+     simplify=FALSE,
+     dat=cat.df)
> best.cv.n<- sapply(best.glm.cvn, FUN=function(x) x$cv.n$delta[1])
```

```
> par(mfrow=c(2,2))
> plot(y=cat.reg.sum$adjr2, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(R[a]^2),
+      pch=3, type="b")
> plot(y=cat.reg.sum$cp, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(C[P]),
+      pch=3, type="b")
> abline(c(1,1), lty=2)
> plot(y=cat.reg.sum$bic, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(BIC),
+      pch=3, type="b")
> plot(y=best.cv.n, x=1:k,
+      xlab="number of covariates",
+      ylab=expression(CV[n]),
+      pch=3, type="b")
```



8. Discuss your findings. In particular, discuss the selection of your best model(s) using the above criteria, BIC, C_P, R_a^2, and $CV_{(n)}$. Please limit your summary to one-half page (or less).
   ANS: Your answers may vary. Given that x_1, x_2 and x_9 occur in the best models chosen by BIC and C_P, and that these 3 inputs occur in all other models as selected by other criteria, then I would choose the model with x_1, x_2 and x_9. It's the most parsimonious best model, too, obviously.

```
> (best.cv<- which.min(best.cv.n))

[1] 4
```

```
> coef(best.glm.cvn[[best.cv]]$fit)

 (Intercept)           x1           x2           x6           x9
 6.823110029 -0.002854864 -0.030868136  0.598591740 -1.227515259

> best.list<- coef(cat.reg, id=c(best.bic, best.cp1, best.cp2, best.adjr2, best.cv)
> names(best.list)<- c("bic","cp1","cp2","adjr2","cvn")
> best.list

$bic
 (Intercept)           x1           x2           x9
 5.711169486 -0.002148421 -0.030582445 -0.598566825

$cp1
 (Intercept)           x1           x2           x6           x9
 6.823110029 -0.002854864 -0.030868136  0.598591740 -1.227515259

$cp2
 (Intercept)           x1           x2           x9
 5.711169486 -0.002148421 -0.030582445 -0.598566825

$adjr2
 (Intercept)           x1           x2           x4           x5
 6.764042285 -0.002787789 -0.033584770 -0.450509338  0.107717534
          x6           x9
 0.434039759 -0.870773297

$cvn
 (Intercept)           x1           x2           x6           x9
 6.823110029 -0.002854864 -0.030868136  0.598591740 -1.227515259
```

9. Suppose we want to predict the (natural log of) the average number of caterpillar nests per tree (y^*) for a new observed input $\mathbf{x}^* = (1150, 20, 2, 4, 15.0, 1.1, 1.1, 6.0, 1.5, 1.3)^t$. Use your best fitted model, chosen via CV(n), above, to compute the predicted value, along with a (nominal) 95% prediction interval for y^*. To do this, use the `predict` function with the `interval='predict'` option, as illustrated in the last code chunk of our note chapter 2. (Incidentally, this assumes a normal linear model with an unbiased fitted model, which we might hope is approximately true; there are other methods to get a prediction interval in our case, but we have not talked about these methods yet.) Show your code/output and summarize briefly your prediction interval. In particular, comment on the validity of the nominal 95% coverage rate and the interval width.

ANS: We predict the unknown (natural log of the) average number of caterpillar nests per tree, associated with input $\mathbf{x}^* = (1150, 20, 2, 4, 15.0, 1.1, 1.1, 6.0, 1.5, 1.3)^t$ (not all of which are used), to be $\widehat{y}^* = \widehat{\mu}(\mathbf{x}^*) = 1.74$. Further, assuming that our final selected model, $\widehat{\boldsymbol{\beta}}$, is approximately unbiased for the unknown "true" function, μ, and assuming that we do not suffer much from overfitting, then we can conclude that y^* is in the inteval $[0.58, 2.90]$ with 95% confidence.

```
> best.cv.lm<- lm(y ~ x1+x2+x6+x9, data=cat.df)
> xstar<- data.frame(x1=1150, x2=20, x3=2, x4=4, x5=15.0,
+                    x6=1.1, x7=1.1, x8=6.0, x9=1.5, x10=1.3)
> predict(best.cv.lm, newdata=xstar, se.fit=TRUE,
+         interval="prediction", level=0.95)

$fit
       fit       lwr      upr
1 1.739832 0.5800144 2.899649

$se.fit
[1] 0.1846782

$df
[1] 28

$residual.scale
[1] 0.5352396
```

```
> detach(package:boot)
> detach(package:leaps)
```

Bibliography

[MR07] Jean-Michel Marin and Christian P. Robert. *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer, New York, 2007.